A Real-time and Non-Intrusive Analyzer for Anomalous Behavior of Computer Networks with Paraconsistent Logic

AVELINO P. PIMENTA JR[1]
JAIR MINORO ABE[2]
SANDRA CRISTINA COSTA PRADO[3]

Paulista University, Graduate Program in Production Engineering
R. Dr. Bacelar, 1212, 04026-002 São Paulo, Brazil
[1]appimenta@gmail.com
[2]jairabe@uol.com.br
[3]sandra.prado01@fatec.sp.gov.br

Abstract - Anomalous behavior in computer networks is always a challenge when a great number of connected devices is considered, as well as the issue of privacy when direct access to the equipment is necessary. Computer viruses, intrusion tools, or malicious programs can produce unexpected changes in a computer network, due to the unusual behavior, if compared to normal operating situations. Therefore, searching for anomalies in the network, which can be understood as situations of inconsistency in several measures of performance of the analyzed devices, should be considered. A possible approach in such situations is the use of a non-classical logic. The Paraconsistent Logic is one of these and it has been increasingly used in several areas for its flexibility and ease. It can determine several additional states, beyond true and false, to deal with situations of inconsistency. This project proposes a new alternative in the search for anomalies in computer networks with Paraconsistent Logic through a non-intrusive approach.

Keywords*:* Paraconsistent Logic, computer networks, inconsistency, anomalous behavior.

(Received November 6st, 2018/ Accepted December 5st, 2018)

1 INTRODUCTION

The purpose of this project is to search for situations that are considered anomalous in a computer network, using a non-intrusive and systemic approach. It is non-intrusive because there is no need for direct access to the network devices, since it uses only the operating attributes available in the router logs. It is systemic because the operation of all network devices is considered to verify the occurrence of inconsistencies in the network. Once these situations that require attention are detected, it is possible to apply the necessary corrections.

Computer networks currently constitute the main form of transmitting data and services. Therefore, the task of monitoring the information has become a key factor in technology sectors [18] and is part of the backbone of information technology in various educational institutions and companies. A computer network consists of several connected hosts, which can be represented by a desktop, a laptop, a smartphone, wearable devices, biomedical sensors, among others [29][34]. In such heterogeneous client environment, efficient content adaptation and delivery services are becoming a major requirement for the new Internet service infrastructure [9].

By its very nature of decentralization and heterogeneity, it is often difficult to determine, in a feasible time, when an anomalous behavior occurs. Errors are often caused by anomalies, which can be understood as unexpected behaviors in a network [13][8]. Often, the discovery of one (or more) spot of failure occurs late, and its correction may be difficult, expensive, and impractical in some cases. Furthermore, with the growing number of World Wide Web users, the need for satisfactory performance becomes increasingly relevant [7]. Thus, early detection of failure spots can certainly bring benefits and avoid significant losses in corporations.

In critical systems, data loss due to malfunction of the network is not an option. Often, the lost information is no longer available for recovery, since the backup policies are often poorly implemented or even inexistent, and it also applies for great corporations. These situations can lead to financial losses [17], and frequently the operating costs arising from data loss cannot be estimated. However, it is important to emphasize that many reported anomalies turn out to be false, reflecting an unusual, although benign behavior [16]. In addition, the establishment of a set of criteria should be done in such a way as to avoid false positives [14], of which developments may lead to several problems, including those of a legal nature.

This paper is organized as follows: in Section 2, there is a comparison of the evaluated project with existing anomaly detection models is discussed. In Section 3, are presented the network attributes used in the learning process are presented. In Section 4, basic concepts of Paraconsistent Logic are introduced. Section 5 discusses the development of the analyzer, considering the network attributes and Paraconsistent Logic. The results and discussion are presented in Section 6, and the conclusion in Section 7.

2 Comparison with existing models

In order to compare this work with existing models, it is necessary to present the materials and tools used in the developed project. The information source may be acquired from different network devices, such as routers, proxies, or switches. In this case, the learning process gathers attribute values from the Squid proxy logs of a customized router. Each workday log varies in size, ranging from 60000 to 150000 records, representing an external resource request by a given network host.

The attributes have been normalized for each host that composes the network, considering the analyzed range. Next, concepts of the Paraconsistent Logic were applied for each piece of equipment and the favorable and contrary evidences of the attributes were determined. With the aid of a data traffic analyzer, it has been possible to determine the network hosts behavior within a specific time interval. Finally, an overall analysis was achieved considering the various possible logical states contemplated by the Paraconsistent Logic.

Unlike the anomaly detection models proposed by [33] and [30], which rely on simulated data to emulate a real network environment and the synthetic generation of anomalies, this project employs continuously gathered data from the operation of a real operating network for the learning process. Another difference from [23] is that it uses Digital Signature of Network Segment using Flow Analysis (DSNSF), which establishes a profile for the normal behavior of a network segment by considering the history of its movement. A possible problem: when a real-time system is not considered for this kind of task, any changes in the network layout or in its availability may impact the analyzer learning process, since the history may not represent the actual state of the network.

Another significant difference from the work of [11] is that, in it, data traffic was used as an analytical measure, without distinction of individual attributes that could represent different operating situations of the network. In this project, the network attributes are treated individually.

3 Network Attributes

As mentioned by [20]: “The amount of information that travels across the Internet has increased dramatically in the past few decades because of the huge growth in the number of Internet users”. Also, the growing number of services and applications, as well as the many advances in information technology, make networks and information systems essential for the survival of all educational enterprises, organizations, and institutions [22]. The growth of the global computer network also leads to an increase in the complexity of its infrastructure. Thus, the classical methods of network analysis may not be the most adequate ones for this scenario [12].

Therefore, responsive service plays a critical role in determining end-user satisfaction, and preserving network stability is important to ensure that services are not disrupted [31]. Network infrastructure needs to be constantly improving to satisfy QoS (Quality of Service) user demands, including both technology aspects (e.g. the fastest links, proxies, and servers) and related software [10]. An important issue to be considered is reliability, which can be understood as the ability of the computer network to successfully transmit data from a specific source to a destination [19], and a performance index to evaluate the capability of a computer network [31]. The maintenance of this feature is a constant challenge, since failures are inevitable in computer networks, and therefore their immediate detection and isolation are necessary [15].

Computer networks use routers to communicate with each other. As mentioned by [32], “Routing is the process of sending data packets from the host of origin to the destination host, which is performed by the routers”. Generally, such devices are called gateways, which operate in the man-in-the-middle function. Many other features may be added to a router, such as access control, firewall, or bandwidth managers.

For the establishment of network communication, there must always be a request from the "client" side. It is a typical protocol of request-response, which controls the data transfer between server and client (such as a web browser) [28].

This request, when answered by the "server" side, triggers a corresponding response. Proxy servers are designed with three goals: decrease network traffic, reduce user (client) perceived lag, and reduce loads on the origin servers [26]. Every request from the client passes through the proxy server, which in turn may or may not modify the client's request based on its implementation mechanism [3].

Some elements may be interesting for the packet traffic analysis, such as the logical address associated with the request for the resource, request time, response time, type of result obtained, amount of response data in the transaction, and destination request [27]. As the gateway or proxy forward packets to other networks, it is possible to audit data traffic information. In this project, this information was referred to as attributes.

There are several analyzable attributes with different importance levels. In this case, the following ones were considered:

Trafficked Data (D)	Response Time (RT)	Requisitions (R)	Errors (E)

When a resource is requested, the responsible agent for locating and searching the Internet is the gateway, which can be represented by a router or a proxy. When receiving this information, it can be registered in logs, which are plain text files that record each of the performed operations with their attribute values.

The first attribute (trafficked data) is measured in bytes and corresponds to the volume of information that was requested by a given network device in each interval. The second attribute (response time) is measured in milliseconds and corresponds to the total response time to obtain the requested resources in each interval. The third attribute (requisitions) corresponds to the number of requests made by a network device in each interval. The fourth attribute (errors) corresponds to the number of zero-byte responses received after a request, which can be translated as an error or problem in the location of a requested resource. As mentioned before, other attributes could be used for this project. However, these have been considered the most significant ones.

4 The Paraconsistent Logic

High levels of uncertainty and unpredictability are an important issue when monitoring computer networks. The argument for this assertion is based on the principle that user actions are presented as random elements [6]. Therefore, the use of a non-classical logic becomes an option. The Paraconsistent Logic can be a viable technique to search for indications of problems, during the normal operation of the network or by intentional elements [24] [25]. In the latter case, it can be caused by misuse or malicious software [21].

According to [2]: "The atomic formulas of the Paraconsistent Logic are the type p(μ, λ), where (μ, λ) ∈ [0, 1]^2 ([0, 1] is the real unit interval) and p denotes a propositional variable". Therefore, among several readings, p(μ, λ) can be intuitively read: "It is assumed that the favorable evidence of p is μ, and the contrary evidence of p is λ". Thus, we have, for instance, the following particular readings:

- $p_{(1.0,\ 0.0)}$ can be read as a true proposition
- $p_{(0.0,\ 1.0)}$ as false
- $p_{(1.0,\ 1.0)}$ as inconsistent
- $p_{(0.0,\ 0.0)}$ as paracomplete, and
- $p_{(0.5,\ 0.5\ )}$ as an indefinite proposition

The uncertainty and certainty degrees associated to (μ, λ) are defined [1][4]:

- Uncertainty Degree: $G_{un}(\mu, \lambda) = \mu + \lambda - 1$ ($0 \leq \mu, \lambda \leq 1$);
- Certainty Degree: $G_{ce}(\mu, \lambda) = \mu - \lambda$ ($0 \leq \mu, \lambda \leq 1$);

An order relation is defined on $[0, 1]^2$: $(\mu_1, \lambda_1) \leq (\mu_2\ \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ and $\lambda_2 \leq \lambda_{1,}$ forming a lattice which is symbolized by τ.

With the degree of certainty and uncertainty, one can determine the following 12 output states, shown in Table 1:

Table 1: Extreme and non-extreme states

Extreme states	Symbol	Non-extreme states	Symbol
True	V	Quasi-true tending to Inconsistent	QV→T
False	F	Quasi-true tending to Paracomplete	QV→⊥
Inconsistent	T	Quasi-false tending to Inconsistent	QF→T
Paracomplete	⊥	Quasi-false tending to Paracomplete	QF→⊥
		Quasi-inconsistent tending to True	QT→V
		Quasi-inconsistent tending to False	QT→F
		Quasi-paracomplete tending to True	Q⊥→V
		Quasi-paracomplete tending to False	Q⊥→F

Extreme and non-extreme states are shown in Figure 1:

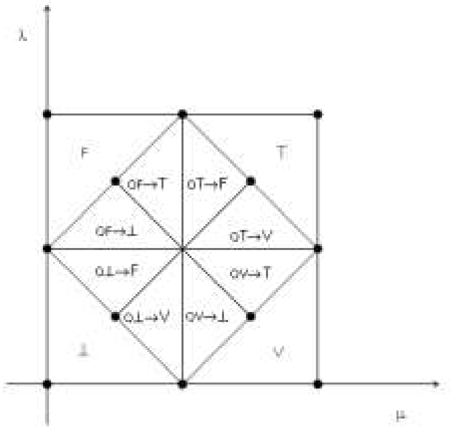

Figure 1: Extreme and non-extreme states of the Lattice τ

In Figure 2, the states, along with certainty and uncertainty degrees, are shown, as well as the control values.

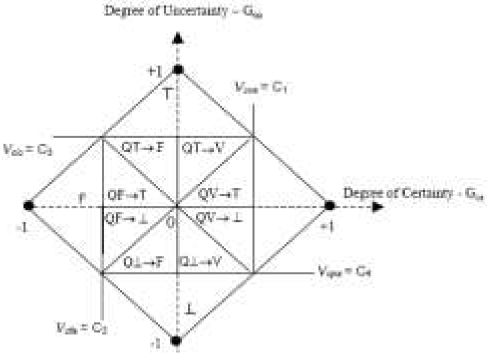

Figure 2: Certainty / Uncertainty degrees with decision states of the Lattice τ

5 Development of the analyzer

For the development of the data traffic analyzer for detecting anomalous behavior in computer networks, it was necessary to transform the data from the Squid log, which was originally formatted in plain text files, to a relational database system. The first step in obtaining the data has been the conversion of each of the tabulated fields from the text file to the CSV (comma-separated values) format, which can be done with common spreadsheet applications. From the newly obtained file, it has been possible to generate the relational database. The Hibernate framework has been used to perform object-relational mapping (ORM), in which the main objective is to reduce the complexity involved in the development of applications that need to interact with relational databases [5]. In this case, the database is converted into objects and can be accessed without the need of explicit SQL calls, thus making native calls. The developed analyzer carries out the monitoring in two distinct stages, defined as follows:

- Systemic evaluation and detection of critical intervals
- Specific detection of network anomalies at critical intervals

5.1 Systemic evaluation and detection of critical intervals

Considering that it is not known at what time a host can perform a non-compatible behavior with normal patterns, the first step is to perform a comprehensive assessment of the network operation. The initial interval was set to 15 minutes, up to the limit of 48 hours of continuous operation, limited only to the available processing capacity. Initially, it has been observed that inferior values (less than 15 minutes) were not sufficient to guarantee an acceptable learning process of the system to generate reliable results, considering that the data were still not sufficiently representative. Conversely, an interval of more than 48 hours generated a negative impact in terms of performance. Given the performance needs, 30 minutes intervals were used in this project.

At this stage, there is still no hint of the hosts that may be potential problem generators, since the emphasis is on the detection of one or more critical intervals in which anomalous behaviors may be occurring. As it is possible to be verified, the **Evidences** comprise the fields of the accumulated values of each network attribute acquired from the **Requisitions**, as well as their normalized values. All attributes are accounted, whether by accumulation or counting. Inactive hosts are not considered, since their attributes without values, if computed, would end up undesirably influencing the process of normalization of values. The attributes acquisition scheme of the critical intervals is represented in Figure 3:

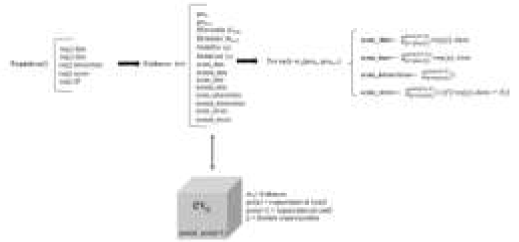

Figure 3: Acquisition scheme of the critical intervals for the Evidences

Initially, a List of the Evidences is generated, based on a given interval. Thus, it is possible to obtain the absolute values of the attributes under analysis. The next step is to normalize the values from the generated List and determine the Favorable (μ) and Contrary (λ) Evidences, as well as the Certainty (G_{ce}) and Uncertainty Degrees (G_{cu}), of the various time intervals, as presented in Figure 4:

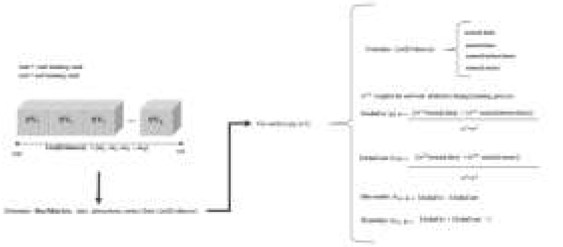

Figure 4: Favorable (μ) and Contrary (λ) Evidences and the Certainty (G_{ce}) and Uncertainty Degrees (G_{un}) for the List of Evidences

A possible approach may be to consider one or more intervals with extreme values, according to the concepts of Paraconsistent Logic. With the result of the interval, the next step is the specific search for the anomalous behavior host.

5.2 Specific detection of network anomalies at critical intervals.

Once the interval has been determined, the first step is to acquire all network attributes for each of the network hosts within that interval. Inactive hosts with attributes without values that, if computed, would undesirably inflate in the process of normalization of values, were not considered. This scheme is presented in Figure 5:

Figure 5: Acquisition scheme of the network hosts for the Host Attributes

Initially, a List of the Host Attributes is generated based on that given interval. The next step is to normalize the values from the List and determine the Favorable (μ) and Contrary (λ) Evidences, as well as the Certainty (G_{ce}) and Uncertainty Degrees (G_{cu}) of the networks hosts, as presented in Figure 6:

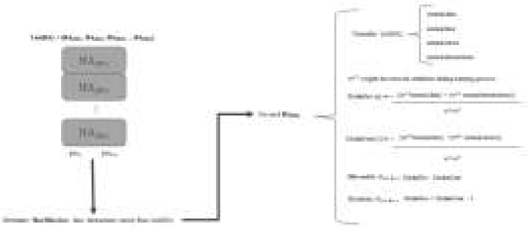

Figure 6: Favorable (μ) and Contrary (λ) Evidences and the Certainty (G_{ce}) and Uncertainty Degrees (G_{un}) for the List of Host Attributes

This part of the process should be repeated as many times as necessary, with a gradual decrease in the search intervals, until the expert can verify a clear and undoubted scenario, that is, the host with anomalous behavior.

5.3 Search for anomalous behavior in time intervals

The first step is to determine the Favorable (μ) and Contrary (λ) evidences of the attributes in the considered intervals. This range is parameterizable, and therefore it can be adjusted according to the presented scenario. For example, in situations where data traffic is considered low, it may be worthwhile increasing the time interval to be analyzed. This would lead to more significant sampling of the object to be analyzed. If data traffic is heavy, this range may be decreased so as not to compromise analyzer performance.

The analyzed scenario uses IPv4 addresses and the lease of the addresses is set up randomly to the requesting devices. However, only active hosts have been considered to calculate the evidences. The analysis of the attributes has been made from 8:00 to 8:29, 8:30 to 9:00, and subsequently until 23:00, for determining the Favorable (μ) and Contrary (λ) evidences of a workday. Hence, several scenarios of network traffic could be evaluated.

For each of the attributes, the normalization process of the values in the intervals between 0 and 1 was applied. This process is necessary for the determination of the Favorable (μ) and Contrary (λ) evidences of the attributes.

It is possible to create three stacked bar graphs that represent the three intervals of the network operating day. In Figure 7, it can be observed that the largest sum of all the normalized attributes is less than 1. Although it is not possible to point out any abnormality in the network, considering that the values are still insufficient and not representative, when it is noted that the "Error" attribute could not even be quantified, the probability of functioning within normality patterns is higher.

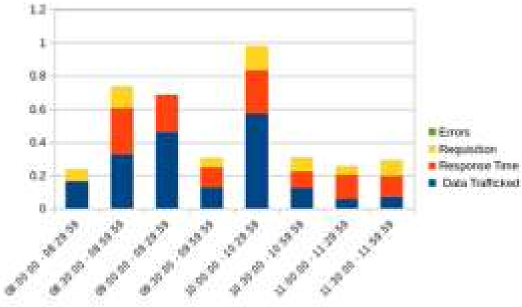

Figure 7: Bar graph representing the network operating from 08:00 – 11:59

Figure 8 presents a similar behavior to Figure 7 until approximately 16:00, when the sum of the normalized attributes abruptly exceeds the value 3, and this behavior prevails until 16:59. This change particularly draws attention, since usually in this interval there are few connected users, and, therefore, little use of the network capacity. In addition, the intervals from 16:00 to 16:29 and 16:30 to 16:59 have significant error rates, which may require a more careful analysis in the search for anomalous events.

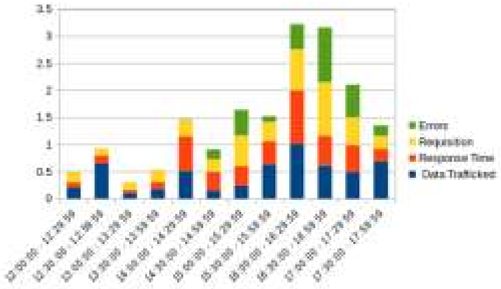

Figure 8: Bar graph representing the network operating from 12:00 – 17:59

In Figure 9, which covers the interval of the most intense use of the network, it is possible to observe that except for the interval from 22:30 – 22:59, the sum of the normalized attributes does not even exceed the value of 1.5. Although the last interval presents a sum that slightly exceeds the value 2, it is still lower than those observed in Figure 8.

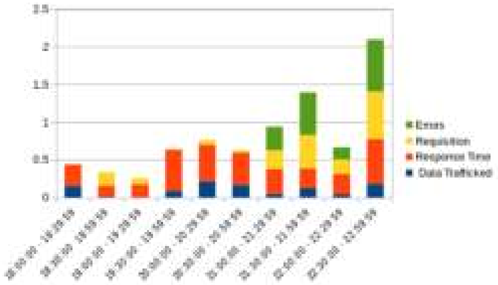

Figure 9: Bar graph representing the network operating from 18:00 – 22:59

Considering that the search for anomalies may be simpler in extreme situations, a good choice to start tracing would be the intervals indicated in Figure 8.

Among the analyzed attributes, a computer network that can forward a significant amount of data satisfactorily, answering to the user's requests, is desirable. Therefore, the attributes Trafficked Data (D) and Requisitions (R) can be considered Favorable evidences.

On the other hand, it is important that the response time is as low as possible in order to obtain higher system usability and user satisfaction. A small number of errors is also desired, which implies fewer retransmissions. High values of the two attributes are not desirable. Therefore, the attributes Response Time (RT) and Errors (E) can be considered Contrary (λ) evidences.

For each analyzed interval, the Favorable (μ) and Contrary (λ) evidences were determined, considering the mentioned attributes, with the following formulas:

$$\mu = (w^1D + w^2R) / (w^1+w^2)$$
$$\lambda = (w^3RT + w^4E) / (w^3+w^4)$$

The next step was to search for one of the time intervals that may represent a significantly anomalous network operation. Among the candidate intervals for analysis, there was a high degree of uncertainty between 16:00 and 16:29. In fact, the G_{ce} was calculated at 0.14512861 and the G_{un} at 0.60824025, revealing a significant inconsistency profile.

With a "divide-to-conquer" strategy, a new analysis was carried out, considering only the interval between 16:00 and 16:30, but with a 10-minute variation. The following values were obtained, with the respective G_{ce} and $G_{un,}$ according to Table 2:

Table 2: Certainty / Uncertainty degrees of the attributes from 16:00 to 16:30

Time interval	Certainty Degree G_{ce}	Uncertainty Degree G_{un}
16:00:00 - 16:09:59	-0.48813787	-0.48813784
16:10:00 - 16:19:59	0.32670146	-0.67150617
16:20:00 - 16:29:59	0.4484431	0.5515568

With the new analysis, another source of inconsistency has been obtained from the interval between 16:20 and 16:28 hours. A new analysis was performed, with another time reduction, 2 minutes, as shown in Table 3:

Table 3: Certainty / Cncertainty degrees of the attributes from 16:20 to 16:30

Time interval	Certainty degree G_{ce}	Uncertainty Degree G_{un}
16:20:00 - 16:21:59	0.23057377	0.2281071
16:22:00 - 16:23:59	0.5198167	-0.39736778
16:24:00 - 16:25:59	-0.49263892	-0.1695013
16:26:00 - 16:27:59	0.0058194995	0.3685069
16:28:00 - 16:29:59	0.08172488	-0.9182751

Among the intervals, an inconsistency between 16:26 and 16:28 was observed. A final analysis, with a variation of 1 minute, was performed, with the following result shown in Table 4:

Table 4: Certainty / Uncertainty degrees of the attributes from 16:26 to 16:28

Time interval	Certainty Degree G_{ce}	Uncertainty Degree G_{un}
16:26:00 - 16:26:59	0	-1

16: 27:00 - 16:27:59	0	1

Among the two intervals, the one that presented the highest level of inconsistency was the one between 16:27 and 16:28. Thus, a 1-minute interval has been determined, in which the potential problem-generating equipment(s) in the network could be tracked. From a 15-hour operating scenario, it has been possible to reduce the scope of the analysis to only 1 minute.

5.4 Search for anomalous behavior of the network hosts

Based on the time interval, between 16:27 and 16:28, a specific search has been held to locate one or more hosts that might be responsible for the anomalous behavior in the computer network. Each of the operating network hosts within the defined time interval, and identified by its source IP address, had calculated their respective Favorable (μ) and Contrary (λ) evidences for the attributes. The following results were obtained, according to the Table 8:

Table 8: Certainty / Uncertainty degrees of the hosts from 16:27 to 16:28

Host	Certainty degree G_{ce}	Uncertainty Degree G_{un}
Host A	0.0712372	-0.881337
Host B	-0.022222161	0.07037747
Host C	-0.05768591	-0.9401534
Host D	-0.12034002	-0.87494224
Host E	-0.082016654	-0.91380996
Host F	0.049036026	0.049036026
Host G	-0.01788491	-0.97344714
Host H	-0.010851777	-0.97934717
Host I	-0.08030032	-0.77098423
Host J	0.058846354	-0.79822236

Ten candidates were determined from 254 possible sources. Among these candidates, the one that represents the most evident anomalous behavior was Host F.

6 Results and Discussion

From the obtained result, a specific host analysis has been carried out, in which the level of inconsistency has been continuously maintained throughout the analysis. It is important to point out that other devices could also simultaneously have presented problems of inconsistency, also becoming candidates for analysis of this nature. However, only one host presented this type of problem, and it underwent a more accurate verification.

A quantitative analysis of the host attributes was performed on the same day. It has been determined that it was a connected device in the administrative sector of the institution. This host had operated only in the interval between 13:30 and 16:30. Further analysis revealed that the computer was infected with various types of viruses.

Considering the interval, the Favorable (μ) and Contrary (λ) evidences were calculated, according to the following Table 9:

Table 9: Normalized attribute values of the host from 13:30 to 16:30

Time interval	Data	Interactions	Response Time	Errors
13:30:00 13:59:59	0.113430575	0.014455948	0.19705948	0.0053908355
14:00:00 14: 29:59	1	0.120868206	0.5505165	0.0062621143
14:30:00 14:59:59	1	0.01741149	0.92535114	0.0008268449
15:00:00 15:29:59	1	0.023295393	1	0.0010223787
15:30:00 15:59:59	0.035427984	0.04673564	0.65221035	0.0048908954
16:00:00 16:29:59	0.98902106	0.11062907	1	0.12903225

Although the device operation occurred within only three hours, it has been possible to observe significant anomalous behavior.

In the interval from 14:00 to 14:29, from 14:30 to 14:59 and from 15:00 to 15:29, the host had its most intense use of the network bandwidth, although the number of requests was considered small, presenting an inconsistent behavior. In fact, considering the three intervals, it was observed that the last two responded by only 1% and 2% of the interactions, respectively, which clearly represents an unexpected behavior of the device.

In all three intervals, the error rate remained low, which shows that the network continued responding to the requests. In the first interval, the response time corresponds to only 55% of the one observed in the second interval, and immediately rises to 92% and 100% for the last intervals, causing the search for network resources to become significantly slower.

From 15:30 to 15:59, there was a sudden drop in data traffic, which have fallen to approximately 3.5% of the total. Even so, the equipment responded for 65% of the response time, an unexpected behavior since the error rate remained low.

Previously, it had already been determined that the interval between 16:00 and 16:29 comprised the one with the worst performance in the network, and this has been confirmed by the analysis of the attributes. In the specific range, the network almost reached the highest utilization of the network (98% of the trafficked data), considering only 11% of the requests. The response time was the highest among the other hosts, and in this interval the error rate increased to 12%, significantly higher than the others, ranging from 0% to 0.4%.

7 CONCLUSION

From this project, it has been possible not only to determine which host presented a contradictory and unexpected behavior within the network parameters but also the exact moment in which this occurred, within a 1-minute interval. By taking into account that the network operates from 8:00 to 23:00, the reduction of 15 hours of analysis to only 1 minute can be considered a success. This reduction would be even more evident in scenarios with greater number of connected devices, where the prospecting of errors would be even more complex.

In any case, it has been possible to determine that the reason for the malfunction of the host did not go through problems such as congestion or network failure. In fact, it has been possible to observe the so-called "misuse" of the equipment. It has also been possible to verify that the equipment, originally installed for the exclusive use of the administrative sector, was full of unauthorized applications, being heavily used for access to unreliable Internet addresses, which unfortunately brought several of the so-called malicious software into the system. These, on an autonomous basis, sent and received data to the Internet, contributing significantly to the emergence of security breaches in the analyzed network.

It is important to emphasize that the search was not for malfunctioning devices due to physical problems, but rather to anomalous situations generated by computers operating within normal patterns and without apparent failures. Eventually, it has been observed that many of the devices identified as causing the anomalies were infected by malwares. The exact type of malware may be the subject of future projects.

REFERENCES

[1] Abe, J. M., *Paraconsistent Intelligent Based-Systems: New Trends in the Applications of Paraconsistency*. Germany: Springer-Verlag, 2015.

[2] Abe, J. M., Akama, S., and Nakamatsu, K., *Introduction to Annotated Logics - Foundations for Paracomplete and Paraconsistent Reasoning*, 1st ed. Springer International Publishing, 2015.

[3] Agarwal, T. and Leonetti, M. A., “Design and Implementation of an IP based authentication mechanism for Open Source Proxy Servers in Interception Mode,” Feb. 2013.

[4] Akama, S., *Towards Paraconsistent Engineering*. Springer International Publishing, 2016.

[5] Babu, C. and Gunasingh, G., “DESH: Database evaluation system with hibernate ORM framework,” in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 2549–2556.

[6] Ben-Porat, U., Bremler-Barr, A., and Levy, H., “Computer and network performance: Graduating from the ‘age of Innocence,’” *Comput. Networks*, vol. 66, pp. 68–81, 2014.

[7] BenaditP, J. and FrancisF, S., “ScienceDirect Improving the Performance of a Proxy Cache Using Very Fast Decision Tree Classifier,” *Procedia - Procedia Comput. Sci.*, vol. 48, no. 48, pp. 304–312, 2015.

[8] Brighenti, C. and Sanz-Bobi, M. A., “Auto-Regressive Processes Explained by Self-Organized Maps. Application to the Detection of Abnormal Behavior in Industrial Processes,” *IEEE Trans. Neural Networks*, vol. 22, no. 12, pp. 2078–2090, Dec. 2011.

[9] Canali, C., Cardellini, V., and Lancellotti, R., “Content Adaptation Architectures Based on Squid Proxy Server,” *World Wide Web*, vol. 9, no. 1, pp. 63–92, Mar. 2006.

[10] Cárdenas, L. G., Sahuquillo, J., Pont, A., and Gil, J. A., “The Multikey Web Cache Simulator: a Platform for Designing Proxy Cache Management Techniques,” *Parallel, Distrib. Network-Based Process. 2004. Proceedings. 12th Euromicro Conf.*, pp. 390–397, 2004.

[11] Fernandes, G., Pena, E. H. M., Carvalho, L. F., Rodrigues, J. J. P. C., and Proença, M. L., “Statistical, forecasting and metaheuristic techniques for network anomaly detection,” in *Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC ’15*, 2015, pp. 701–707.

[12] Fernandez-Prieto, J. a., Canada-Bago, J., Gadeo-Martos, M. a., and Velasco, J. R., “Optimisation of control parameters for genetic algorithms to test computer networks under realistic traffic loads,” *Appl. Soft Comput. J.*, vol. 12, no. 7, pp. 1875–1883, 2012.

[13] Fidalgo, J. N. and Lopes, J. A., “Load Forecasting Performance Enhancement When Facing Anomalous Events,” *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 408–415, Feb. 2005.

[14] Fossaceca, J. M., Mazzuchi, T. a., and Sarkani, S., “MARK-ELM: Application of a novel Multiple Kernel Learning framework for improving the robustness of Network Intrusion Detection,” *Expert Syst. Appl.*, vol.

42, no. 8, pp. 4062–4080, 2015.

[15] Garshasbi, M. S., “Fault localization based on combines active and passive measurements in computer networks by ant colony optimization,” *Reliab. Eng. Syst. Saf.*, vol. 152, pp. 205–212, Aug. 2016.

[16] Grana, J., Wolpert, D., Neil, J., Xiea, D., Bhattacharya, T., and Bent, R., “A likelihood ratio anomaly detector for identifying within-perimeter computer network attacks,” *J. Netw. Comput. Appl.*, vol. 66, pp. 166–179, May 2016.

[17] Lee, Y.-J., Yeh, Y.-R., and Wang, Y.-C. F., “Anomaly Detection via Online Oversampling Principal Component Analysis,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1460–1470, Jul. 2013.

[18] Lin, Y. K. and Huang, C. F., “Stochastic computer network under accuracy rate constraint from QoS viewpoint,” *Inf. Sci. (Ny).*, vol. 239, pp. 241–252, 2013.

[19] Lin, Y. K. and Yeh, C. T., “Using minimal cuts to optimize network reliability for a stochastic computer network subject to assignment budget,” *Comput. Oper. Res.*, vol. 38, no. 8, pp. 1175–1187, 2011.

[20] Masuda, K., Ishida, S., and Nishi, H., “Cross-site Recommendation Application Based on the Viewing Time and Contents of Webpages Captured by a Network Router,” 2013.

[21] Misra, a. K., Verma, M., and Sharma, A., “Capturing the interplay between malware and anti-malware in a computer network,” *Appl. Math. Comput.*, vol. 229, pp. 340–349, 2014.

[22] Obaidat, M. S., Nicopolitidis, P., and Zarai, F., “Modeling and Simulation of Computer Networks and Systems,” Elsevier, 2015, pp. 187–223.

[23] Pena, E. H. M., Barbon, S., Rodrigues, J. J. P. C., and Proenca, M. L., “Anomaly detection using digital signature of network segment with adaptive ARIMA model and Paraconsistent Logic,” *Proc. - Int. Symp. Comput. Commun.*, 2014.

[24] Pimenta, A. P., Abe, J. M., and de Oliveira, C. C., *An analyzer of computer network logs based on paraconsistent logic*, vol. 460. 2015.

[25] Pimenta Jr, A. P. and Abe, J. M., “Determination of operating parameters and performance analysis of computer networks with Paraconsistent Annotated Evidential Logic Eτ,” *IFIP Adv. Inf. Commun. Technol.*, vol. 1, pp. 1–9, 2016.

[26] Romano, S. and ElAarag, H., “A neural network proxy cache replacement strategy and its implementation in the Squid proxy server,” *Neural Comput. Appl.*, vol. 20, no. 1, pp. 59–78, Sep. 2010.

[27] Rousskov, A. and Soloviev, V., “A performance study of the Squid proxy on HTTP/1.0,” *World Wide Web*, vol. 2, no. 1, pp. 47–67, 1999.

[28] Sysel, M. and Doležal, O., “An Educational HTTP Proxy Server,” *Procedia Eng.*, vol. 69, pp. 128–132, 2014.

[29] Xu, L. Da, He, W., and Li, S., “Internet of Things in Industries: A Survey,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[30] Yaacob, A. H., Tan, I. K. T., Chien, S. F., and Tan, H. K., “ARIMA Based Network Anomaly Detection,” in *2010 Second International Conference on Communication Software and Networks*, 2010, pp. 205–209.

[31] Yeh, C. T. and Fiondella, L., “Optimal redundancy allocation to maximize multi-state computer network reliability subject to correlated failures,” *Reliab. Eng. Syst. Saf.*, 2016.

[32] Zazuli, L. and Mardedi, A., “Developing Computer Network Based on EIGRP Performance Comparison and OSPF,” *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 9. pp. 80–86, 2015.

[33] Zhu, B. and Sastry, S., “Revisit Dynamic ARIMA Based Anomaly Detection,” in *2011 IEEE Third Int’l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int’l Conference on Social Computing*, 2011, pp. 1263–1268.

[34] Zhuming Bi, Li Da Xu, and Chengen Wang, “Internet of Things for Enterprise Systems of Modern Manufacturing,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1537–1546, May 2014.